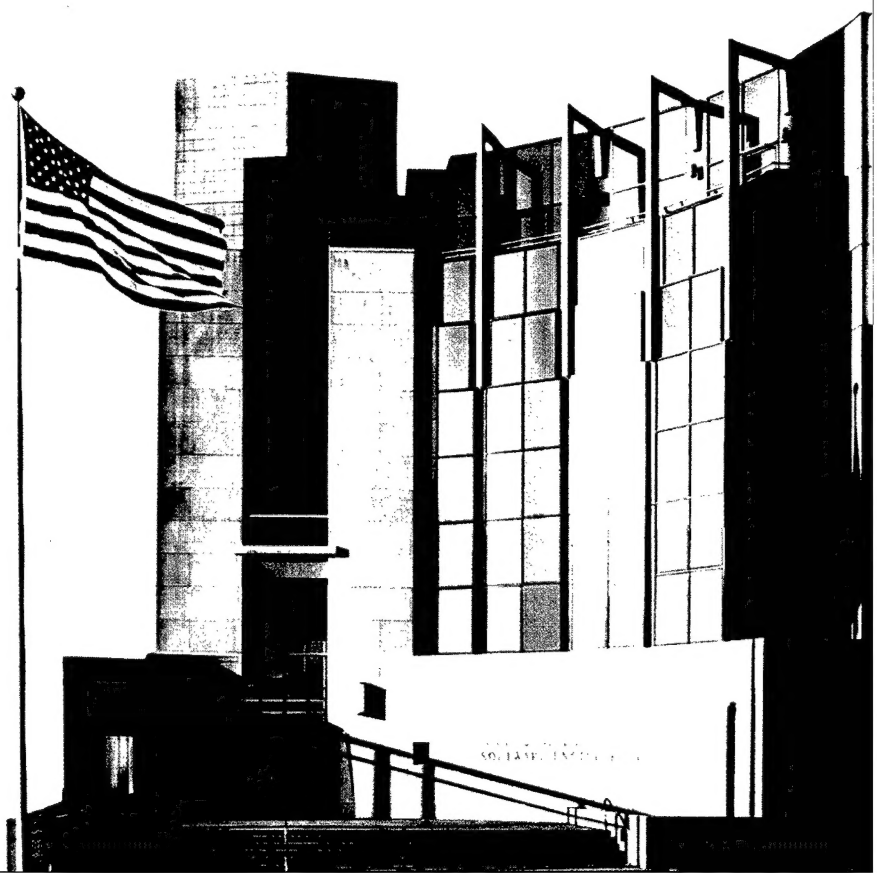
Carnegie Mellon
Software Engineering Institute

Architecture Tradeoff Analyses of C4ISR Products

Mario R. Barbacci
William G. Wood

June 1999

TECHNICAL REPORT
CMU/SEI-99-TR-014
ESC-TR-99-014



Carnegie Mellon University does not discriminate and Carnegie Mellon University is required not to discriminate in admission, employment, or administration of its programs or activities on the basis of race, color, national origin, sex or handicap in violation of Title VI of the Civil Rights Act of 1964, Title IX of the Educational Amendments of 1972 and Section 504 of the Rehabilitation Act of 1973 or other federal, state, or local laws or executive orders.

In addition, Carnegie Mellon University does not discriminate in admission, employment or administration of its programs on the basis of religion, creed, ancestry, belief, age, veteran status, sexual orientation or in violation of federal, state, or local laws or executive orders. However, in the judgment of the Carnegie Mellon Human Relations Commission, the Department of Defense policy of "Don't ask, don't tell, don't pursue" excludes openly gay, lesbian and bisexual students from receiving ROTC scholarships or serving in the military. Nevertheless, all ROTC classes at Carnegie Mellon University are available to all students.

Inquiries concerning application of these statements should be directed to the Provost, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, telephone (412) 268-6684 or the Vice President for Enrollment, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, telephone (412) 268-2056.

Obtain general information about Carnegie Mellon University by calling (412) 268-2000.

Carnegie Mellon
Software Engineering Institute

Pittsburgh, PA 15213-3890

Architecture Tradeoff Analyses of C4ISR Products

CMU/SEI-99-TR-014
ESC-TR-99-014

Mario R. Barbacci
William G. Wood

June 1999

Architecture Tradeoff Analysis Initiative

This work is sponsored by the U.S. Coast Guard. The Software Engineering Institute is a federally funded research and development center sponsored by the U.S. Department of Defense.

This work was created in the performance of Federal Government Contract Number F19628-95-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 52.227-7013.

For information about purchasing paper copies of SEI reports, please visit the publications portion of our Web site (http://www.sei.cmu.edu/publications/pubweb.html).

Table of Contents

List of Figures

List of Tables

Abstract

Early evaluation of the architecture of a system or a product line of systems is a low-cost risk reduction method for determining whether the system(s) will achieve its business and quality goals. The Architecture Tradeoff Analysis Method (ATAM) is an architecture evaluation technique currently evolving at the Software Engineering Institute (SEI). The input to the ATAM consists of a system or product line architecture and the perspectives of stakeholders involved with that system or product line. The output of the ATAM is (1) a collection of scenarios that help specify the context of the system's or product line's use and the product line's evolution, (2) improved architectural documentation (usually), and (3) analysis results (in particular, a set of issues to consider, risks, and potential sensitivity and tradeoff points within the architecture). Currently, there are no generally accepted industry-wide standards for describing a system architecture, and ATAM evaluations are often tailored to the available documentation.

The Architectures Directorate of the C4I Integration Support Activity (CISA), Office of the Assistant Secretary of Defense for Command, Control, Communications, and Intellegence (OASD[C3I]) has defined a framework for architecture development, presentation, and integration to be used across the military services and defense agencies. This framework for Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance (C4ISR) is becoming the required method for describing information systems within the Department of Defense (DoD) and other U.S. Government agencies. This report describes how various C4ISR products can be used in the context of an ATAM evaluation and their relative value for generating quality attribute-specific scenarios required for an ATAM evaluation.

1 The Architecture Tradeoff Analysis Method (ATAM)

The Architecture Tradeoff Analysis Method (ATAM) is an architecture evaluation technique currently evolving at the Software Engineering Institute (SEI). The input to the ATAM consists of a system or product line architecture and the perspectives of stakeholders involved with that system or product line. The ATAM relies on generating use-case and change-case scenarios to assess the architecture. Use cases and change cases are described below:

- *Use cases* represent operational conditions that the architecture should support and demonstrate the effectiveness of the architecture to satisfy these operating conditions.
- *Change cases* represent expected system modifications that may cause architectural changes and demonstrate how efficiently the architecture can be modified.

The ATAM achieves its architecture evaluation for quality-attribute goals by using an understanding of the architectural mechanisms that are used to achieve particular quality goals and the implications of those mechanisms. Each quality attribute is examined from the point of view of three perspectives: what external stimuli causes the architecture to respond or change, what mechanisms are used within the architecture to control the response, and how the response to these stimuli is measured. For performance, for example, the external stimuli are events arriving at the system; the mechanisms are scheduling, concurrency, and resource management; and the measurements are latency or throughput. For modifiability, for example, the external stimuli are changes to the system, the mechanisms for controlling the cost of changes are encapsulation and data indirection, and the measurement is the cost of a collection of changes.

The ATAM consists of a number of steps, starting with a presentation of the business drivers, the architecture, and the identification of architecture styles in the system. This is followed by the generation of scenarios, prioritization of the scenarios, and mapping the prioritized scenarios onto the architecture styles.

The ATAM uses stakeholder perspectives to derive a collection of scenarios giving specific instances for usage, performance requirements or growth, various types of failures, various possible threats, and various likely modifications. The scenarios are used for the evaluators to understand the action of the system or product line. The evaluation results in three different types of output:

1. a collection of "sensitivity" or "tradeoff" points. A sensitivity point is a component or decision made in the architecture that is critical for the achievement of a particular quality attribute. A tradeoff point is a sensitivity point that is sensitive for multiple quality attributes

2. a framework for reasoning about the system or product line. The framework for reasoning about the system or product line may take a variety of forms. It may be the discussion that follows from the exploration of a scenario, it may be a model or a portion of a model and a discussion of how that model might be analyzed when instantiated, or it may be a formula that represents how to calculate a value of a particular quality attribute.

3. a list of issues, or decisions not yet made. The list of issues or decisions not yet made arises from the stage of the life cycle of the evaluation. An architecture represents a collection of decisions. Not all decisions may have been made, even when designing the architecture. Some of these decisions are known to the development team as having not been made and are on a list for further consideration. Others are unknown to the development team and stakeholders.

The only ATAM step affected by the C4ISR[1] Framework products is the scenario generation. The other steps in the evaluation phase are not affected by the C4ISR, since they assume only that some architectural representations will be available on which to map the scenarios.

1. C4ISR is Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance.

2 C4ISR Architecture Framework

The C4ISR Architecture Framework defines three views (operational, system, and technical) and two classes of products (essential and supporting). The three views are defined in the Sections 2.1.1 through 2.1.3 of the framework document and are summarized below:

1. The *operational architecture view* is a description of the tasks and activities, operational elements, and information flows required to accomplish or support a military operation. It contains descriptions (often graphical) of the operational elements, assigned tasks and activities, and information flows required to support the war-fighter. It defines the types of information exchanged, the frequency of exchange, which tasks and activities are supported by the information exchanges, and the nature of information exchanges in detail sufficient to ascertain specific interoperability requirements.

2. The *systems architecture view* is a description, including graphics, of systems and interconnections providing for, or supporting, war-fighting functions. For a domain, the systems architecture view shows how multiple systems link and interoperate, and it may describe the internal construction and operations of particular systems within the architecture. For the individual system, the systems architecture view includes the physical connection, location, and identification of key nodes (including materiel item nodes), circuits, networks, war-fighting platforms, etc., and it specifies system and component performance parameters (e.g., mean time between failure, maintainability, availability). The systems architecture view associates physical resources and their performance attributes to the operational view and its requirements per standards defined in the technical architecture.

3. The *technical architecture view* is the minimal set of rules governing the arrangement, interaction, and interdependence of system parts or elements, whose purpose is to ensure that a conforming system satisfies a specified set of requirements. The technical architecture view provides the technical systems-implementation guidelines upon which engineering specifications are based, common building blocks are established, and product lines are developed. The technical architecture view includes a collection of the technical standards, conventions, rules, and criteria organized into profile(s) that govern system services, interfaces, and relationships for particular systems architecture views and that relate to particular operational views.

The framework specifies that the essential products must be supplied, and different systems could require different supporting products depending on the context. However, it does not provide more specific guidance regarding what supporting products might be needed:

The decision of which products to build, beyond the essential set, must be made based on the issues the architecture is intended to explore and the resulting characteristics that the architecture must capture. A given architecture may contain all of the supporting products, a selected subset, or none of the supporting products. For example, an architecture that is to be used in business process reengineering should include an Activity Model; an architecture that is to be used in examining technology insertion and achievable states of "to-be" capabilities should include a System Technology Forecast [C4ISR Architecture Framework Version 2, Section 3.1.2.2].

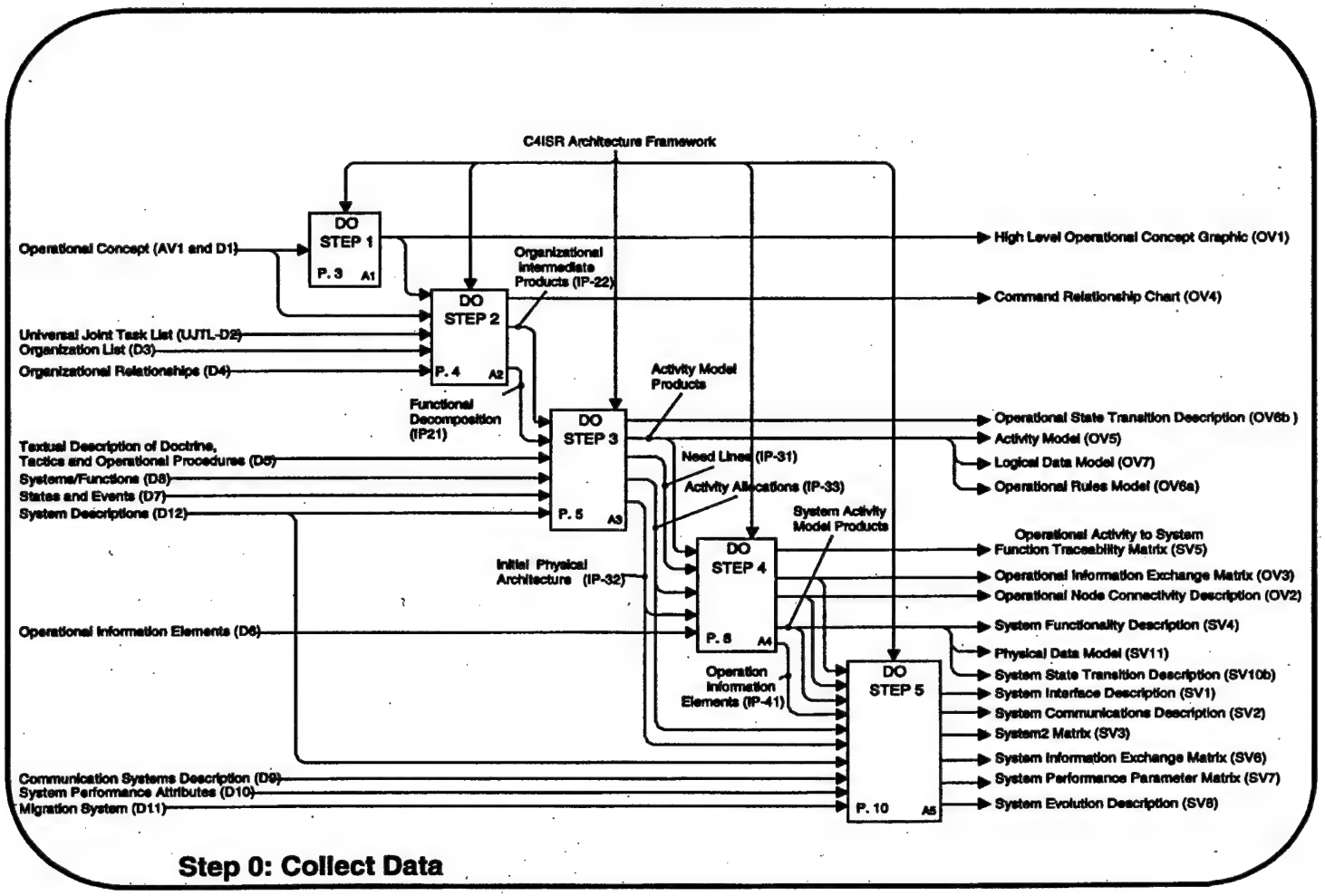
The situation is further complicated because the framework does not provide a process for generating the products. Thus, an organization developing an architecture that is compliant with the C4ISR Framework could be faced with an unbounded amount of effort.

The Systems Architectures Laboratory at George Mason University has developed a C4ISR Architecture Framework Implementation that includes a process for generating the essential and supporting products for the operational and systems architecture views. The process takes a systems engineering point of view and identifies relationships between the products that lead to a six-step process:

1. STEP 0: Problem definition and collection of domain information
2. STEP 1; Operational concept and requirements
3. STEP 2: Functions and organizations
4. STEP 3: Activity model, logical data model, need lines, system nodes, system elements and functions, and task allocation
5. STEP 4: Operational information elements and exchanges, system functionality description, physical data model
6. STEP 5: System information elements and exchanges, local area networks/wide area networks (LAN/WANs), system interface descriptions, system performance

This process is shown in Figure 2-1, which was reproduced from the AFCEA[1] Course 503E, Chapter 3, Page 4.

1. AFCEA is the Armed Forces Communications and Electronics Association.



v. E - 503

Figure 2-1: The Six-Step Process

The process is complex because there is tight coupling among products, and the number of relationships increase dramatically with the layers of representation in the activity model (OV-2) and the system functionality description (SV-4). However, the process produces products in all six steps, allowing for a continuous review and evaluation of the architecture description process.[1]

This report does not deal with product generation; we assume that some collection of products has been developed. Instead, we focus on the task of evaluating the architecture described by the products. The preceding discourse simply points out that the effort required to generate the products is not trivial; therefore, it is important to choose wisely. As we shall see, depending on the combination of products available at the time of the evaluation, different types of attribute-specific scenarios can be generated. Thus, the choice of products could be driven by the quality attributes of interest.

One reason for the complexity of relationships between the operational and system views is that the operational activities can be developed quite separately from the system's functions. Hence a general activity can involve a number of system functions, and a system function can perform parts of many activities, leading to the complex relationships. The complexity can be reduced by carefully choosing the activities to correspond to the system functions.

The techniques used in the C4ISR Framework derive from the structured analysis tradition. Over the past 10 years, object-oriented methodologies have become popular, but object orientation forces the developers to consider only relationships between objects and removes the "activity" versus "function" split. The Unified Modeling Language (UML) supports the development of object-oriented views and is becoming an industry standard. UML products, however, are not interchangeable with C4ISR products since they are developed with a different mindset (structure analysis versus object orientation). The use of similar names for different concepts in the two traditions makes it difficult to know what is meant by a particular term.

1. Alexander Levis and his team at the Systems Architectures Laboratory, George Mason University have built an executable model of the C4ISR products. Such an executable model could be used to validate the C4ISR products, by developing, executing, and analyzing many test cases.

3 Linking the C4ISR Architecture Framework and the ATAM

The C4ISR Framework concentrates on view and product definitions, and it does not require use cases or change cases to clarify and assess the effectiveness of the architecture. The ATAM Evaluation phase relies on generating use-case and change-case scenarios to analyze the architectures. It is not immediately obvious which types of use cases and change cases can or can not be analyzed against which architectural views and products. This is further complicated by the fact that the framework allows the products to be enhanced by annotations or by innovative usage. Products may even be replaced by alternative products that are deemed to be equivalent to the C4ISR Framework products.

3.1 C4ISR Products as Potential Sources of Scenarios

We reviewed the descriptions of the C4ISR products in Section 4 of the framework and the product attribute tables defined in Appendix A of the C4ISR Architecture Framework Version 2, and we determined how could they be used to generate scenarios for use cases and change cases associated with performance, dependability, security, modifiability, and interoperability. In doing so, we assumed a strict interpretation of the appendices.[1] Our detailed observations are listed in Appendix A and are summarized in Table 3-1 and Table 3-2. The tables indicates what attribute-specific information is contained in the product, information which could be used to generate attribute-specific scenarios.

1. Obviously more could be done with any view that is made richer by annotations or innovation.

Possible sources of attribute-specific scenarios from individual products	Performance	Dependability	Security	Modifiability/ Interoperability
Overview and Summary Information (AV-1)				
Integrated Dictionary (AV-2)				
High-Level Operational Concept Graphic (OV-1)	throughput latency	points of failure availability of assets	exposure intersection spoofing	
Operational Node Connectivity Description (OV-2)				
Operational Information Exchange Matrix (OV-3)	message size media frequency of exchanges timeliness throughput	redundancy	classification of messages	protocols formats LISI levels[a]
System Interface Description (SV-1)	throughput channel capacity latency	redundancy of components/ functions	encryptions	standards protocols components contained functions performed
Technical Architecture Profile (TV-1)				protocols formats

Table 3-1: Summary of Essential Products

a. LISI is levels of information systems interoperability.

Possible sources of attribute-specific scenarios from individual products	Performance	Dependability	Security	Modifiability/ Interoperability
Command Relationships Chart (OV-4)		backups		
Activity Model (OV-5)		flow of activities		
Operational Activity Sequence and Timing Descriptions (OV-6a, 6b, and 6c)	alternative paths throughput latency	points of failure		
Logical Data Model (OV-7)	message size	value faults		
Systems Communications Description (SV-2)	WAN/LAN characteristics steps in path	fault rates redundancy	classification encryption spoofing	
Systems^2 Matrix (SV-3)			classification	functionality evolution
Systems Functionality Description (SV-4)				
Operational Activity to System Function Traceability Matrix (SV-5)				
System Information Exchange Matrix (SV-6)	frequency timeliness throughput		classification	media format
System Performance Parameters Matrix (SV-7)	data rates throughput capacity initialization/ restart time response time	MTTF for hardware/ software		
System Evolution Description (SV-8)				upgraded functionality
System Technology Forecast (SV-9)				upgraded capabilities
System Activity Sequence and Timing Descriptions (SV-10a, 10b, and 10c)	execution paths event times	points of failure	points of attack	removing constraints on functionality
Physical Data Model (SV-11)	message size	file system properties		
Standards Technology Forecast (TV-2)				new standards

Table 3-2: Summary of Supporting Products

3.2 C4ISR Products and ATAM Screening Questions

During scenario gathering, the evaluators use screening questions to elicit use-case and change-case scenarios from the stakeholders, and they prioritize these cases to determine which ones will be used for analysis. The high-priority scenarios will be a mixture of scenarios, some that are strictly operational and some that are system oriented.

The screening questions in an ATAM are targeted at what is equivalent to C4ISR system-level views rather than C4ISR operational views. This section describes how these questions could be extended to include questions targeted at the C4ISR operational views. Appendix B lists example screening questions that are appropriate for operational views, and Appendix C lists examples of actual ATAM scenarios.

There are two types of C4ISR operational view screening questions: those concerning the operational environment, which help us to derive an operational context for the developing detailed use-case scenarios, and those concerned with the logistics (support) environment, which help us to derive a logistics context for detailed change-case scenarios.

3.2.1 Operational Environment

The intent of the screening questions (contained in Appendix B.1) is to move from a very wide-scale operational problem set to a much smaller scale problem set that is likely to expose the architectural weaknesses and strengths. They should result in focusing the use cases.

3.2.1.1 Mission and Geography

These questions (contained in Appendix B.1.1) help us to focus on those issues that are stressful to the operational architecture and those that are difficult to accomplish. Information uncovered by these questions can also create a general background environmental "workload" that represents those conditions. A workload could, for example, indicate the number of vessels and aircraft within the area of interest (AOI), describe how an operational picture is achieved and distributed to the platforms and command centers involved with the mission, and show the acceptable latency for upgrades of this picture.

3.2.1.2 Organization and Command Structure

These questions (contained in Appendix B.1.2) help us to establish the command structures involved and how the structures interact to conduct the missions. These questions could lead us to understand which missions could have command-structure interactions that are problematic, or which missions will have command-structure problems when failures arise.

3.2.1.3 Operational Abstract Quality Attributes

The operational environments established by the previous screening questions can now be used to develop abstract scenarios for the appropriate quality attributes, such as availability, performance, security, and interoperability. For each such quality attribute, there are seeding questions (contained in Appendix B.1.3) that will lead to some abstract scenarios. To generate the scenarios, the architect will have to reference some high-level operational views of the system.

3.2.2 Logistics Environment

The screening questions (contained in Appendix B.2) will help us focus the problems with system upgrades and/or logistics. They should lead to an understanding of the strategies involved in these issues and allow us to focus the change-case scenarios appropriately.

3.2.2.1 Platform and Command Center Upgrades

These questions (contained in Appendix B.2.1) help us to establish the ground rules under which platform and/or command-center upgrades will be made. For example, how are classes of cutters and aircraft upgraded? Are cutters upgraded during routine maintenance operations, or can certain upgrades be done less intrusively? Are all aircraft of a certain class removed from service, their crews retrained, and then all restored to service? The screening questions should lead to a basic understanding of these ground rules for platform and command-center upgrades.

A related issue is that of maintaining platform configurations. How much variety is allowed between platforms in the same and different operational areas?

3.2.3 Evolution Strategy

Many of the infrastructure components within certain systems become obsolete in relatively short time periods. There can be a strategy of extending the life of these components beyond its normal range or a strategy of timely replacement of components. With software, the problem is compounded due to the relatively short lifetime of software components. These questions (contained in Appendix B.2.2) will help us to focus the scenarios.

3.2.4 Legacy Systems

Many of the legacy systems will remain until retirement, and in some cases they must interoperate with the new systems being introduced. The screening questions (contained in Appendix B.2.3) in this area should lead to an understanding of the "ground rules" under which these operations will occur.

3.2.5 Logistics Abstract Quality Attributes

The results of the above screening process should allow us to capture the major issues to be considered for the change cases. The questions (contained in Appendix B.2.4) will help us to generate change cases for the quality attributes of modifiability and interoperability. The process here parallels that of the use cases, concerning levels of abstraction and detail in both the change-case scenarios and the view that are referenced.

4 Conclusions and Recommendations

The product summaries in Table 3-1 and Table 3-2 suggest how each product could contribute to the development of quality attribute-specific scenarios by using the ATAM screening questions.

The products are complementary with respect to quality attributes, and different combinations of products could be used to generate different scenarios, depending on the attributes of interest. For example, Table 4-1 show the consolidation of all the essential products, the supporting operational view products, and selected products focused on system evolution. Any number of these combinations could be used as guidance to identify products that are desirable for generating attribute-specific scenarios.

Possible sources of attribute-specific scenarios from product combinations	**Performance**	**Dependability**	**Security**	**Modifiability/ Interoperability**
Essential Products • Overview and Summary Information (AV-1) • Integrated Dictionary (AV-2) • High-Level Operational Concept Graphic (OV-1) • Operational Node Connectivity Description (OV-2) • Operational Information Exchange Matrix (OV-3) • System Interface Description (SV-1) • Technical Architecture Profile (TV-1)	channel capacity message size media frequency of exchanges timeliness throughput latency	redundancy of components/ functions points of failure availability of assets	classification of messages encryptions exposure intersection spoofing	standards protocols formats components contained functions performed formats LISI levels
Supporting Operational View Products • Command Relationships Chart (OV-4) • Activity Model (OV-5) • Operational Activity Sequence and Timing Descriptions (OV-6a, 6b, and 6c) • Logical Data Model (OV-7)	alternative paths throughput latency message size	backups flow of activities points of failure value faults		
Supporting Products Focused on System Evolution • System Evolution Description (SV-8) • System Technology Forecast (SV-9) • Standards Technology Forecast (TV-2)				upgraded functionality upgraded capabilities new standards

Table 4-1: Product Combinations

Both use-case and change-case scenarios can be expressed at the level consistent with the C4ISR operational views or system views. Scenarios expressed at the level of the operational views can be further refined in terms of the more detailed system views, as described below:

- Scenarios at the level of operational views will be expressed in terms of nodes, information flows, and activities. These can be analyzed against the operational views.
- Scenarios at the level of the system views will be expressed in terms of systems, functions, and communication paths. These can be analyzed against the system views.

For example, the following scenarios could be analyzed against operational views:

- A use-case scenario may explore the impact of a node failure on a specific mission in a specific geographical region.
- A change-case scenario might explore the impact of adding a new communications satellite system to a mission.

If the C4ISR operational view products are sufficiently detailed, the scenarios can be analyzed to determine their impact on the mission in terms of the nodes, command structure, activities, and communication paths. However, the scenarios' impact on specific systems or functions cannot be determined at this level. In these cases, the scenarios must be refined in terms of systems and functions and their connectivity, and then explored against the C4ISR system views.

Appendix A: C4ISR Products as Sources of Scenarios

The C4ISR Framework distinguishes between essential and supporting products, and for each product it defines the information to be included in the product. These information tables are captured in the integrated dictionary, which is itself one of the products (AV-2).

In this appendix, we identify items in the information tables that could be used to derive scenarios. This is a literal scan of the product information tables, looking for implicit or explicit references to quality-attribute concerns. Further, we suggest how combining the views could provide coverage across multiple attributes.

A.1 Essential Products

A.1.1 Overview and Summary Information (AV-1)

This view is both a planning guide and documentation upon completion of the development.

It can be used to identify assumptions, constraints, attribute-specific requirements, and scenarios.

A.1.2 Integrated Dictionary (AV-2)

This is a central source for definitions and annotations to the graphical representations.

It can be used to document

- attribute-specific data (e.g., rates, probabilities, impact of changes)
- risks identified during the architecture evaluation

A.1.3 High-Level Operational Concept Graphic (OV-1)

A graphical representation of operations in terms of such things as missions, functions, organizations, and/or asset distribution. It describes weapon types, logical and physical connections, trajectories and targets, and criticality of mission.

It can be used to assign weights to attribute-specific evaluations, sensitivities, and tradeoffs.

Possible scenarios include

- dependability (from points of failure and availability of assets)
- performance (from throughput and latency of communications)
- security (from exposure, interception, and spoofing)

A.1.4 Operational Node Connectivity Description (OV-2)

This describes the operational nodes, the need lines between them, and the characteristics of the information exchanged. Details appear in OV-3.

A.1.5 Operational Information Exchange Matrix (OV-3)

This matrix captures requirements for information exchanges between operational nodes. It adds details to OV-2.

Possible scenarios include

- interoperability (from protocols, formats and LISI levels)
- security (from classification of messages)
- dependability (from redundancy)
- performance (from message size, media, frequency of exchange, timeliness, and throughput data)

A.1.6 System Interface Description (SV-1)

This description links together the operational and systems architecture views by depicting the assignments of specific systems and their interfaces to the nodes and need lines described in OV-2. See SV-2 for communication nodes. See SV-6 for system information elements

Possible scenarios include

- interoperability (from standards and protocols)
- modifiability (from components contained and functions performed by each component)

- dependability (from redundancy of components/functions)
- performance (from throughput, channel capacity, and latency data)
- security (from encryption)

A.1.7 Technical Architecture Profile (TV-1)

This profile provides a time-phased enumeration of the relevant subset of technical standards that apply to the architecture and how they have been or are to be implemented.

Possible scenarios include

- interoperability (from protocols)
- modifiability (from protocols and formats)

A.2 Supporting Products

A.2.1 Command Relationships Chart (OV-4)

This chart illustrates the hierarchy of organizations or resources in an architecture and the relationship among them (e.g., command, control, coordination).

A possible scenario is dependability (from backups).

A.2.2 Activity Model (OV-5)

This model describes the applicable activities associated with the architecture, the data and/or information exchanged between activities, and the data and/or information exchanged with other activities outside the scope of the model (i.e., external interfaces). See OV-2 for operational node and OV-3 for operational information element.

A possible scenario is dependability (from flow of activities).

A.2.3 Operational Activity Sequence and Timing Descriptions (OV-6a, 6b, and 6c)

These descriptions consist of a set of three types of models needed to refine and extend the operational view, to adequately describe the dynamic behavior and performance characteristics of the business processes critical to an architecture. See OV-2 for operational nodes. These three models are described below:

- Operational rules model (OV-6a) — operational rules expressed in a formal language
- Operational state transition description (OV-6b) — detailed timing sequence of activities or work flow in the business process
- Operational event/trace description (OV-6c) — depiction of the dynamic behavior of mission processes (used alone or in conjunction with OV-6b)

Possible scenarios include

- performance (from alternative paths, throughput, and latency)
- dependability (from points of failure)

A.2.4 Logical Data Model (OV-7)

This model describes the data and information that are associated with the information exchanges of the architecture.

Possible scenarios include

- performance (from message size)
- dependability (from value faults)

A.2.5 Systems Communications Description (SV-2)

This description represents the specific pathways or networks of the communications systems and the details of their configurations through which the physical node and systems interface. See SV-1 for systems nodes, systems, and links. See OV-2 for need lines and operational nodes.

Possible scenarios include

- performance (from WAN/LAN characteristics and steps in the path)
- dependability (from fault rates and redundancy)
- security (from classification, encryption and spoofing)

A.2.6 Systems^2 Matrix (SV-3)

This is a description of the system-to-system relationships identified in the various types (e.g., internodal and intranodal) in SV-1. See SV-6 for system information elements.

Possible scenarios include

- interoperability (from existing/planned/potential/deactivated functionality)
- security (from classification of data)

A.2.7 Systems Functionality Description (SV-4)

This describes the flow of data among system functions, and the relationship between systems or system functions and activities at nodes. See SV-1 for system function and system node. See SV-6 for system information element.

This is not a likely source of scenarios.

A.2.8 Operational Activity to System Function Traceability Matrix (SV-5)

This matrix helps to link the operational and systems architecture views by depicting the "many-to-many" mappings of operational activities to systems functions. See SV-1 for system function. See OV-5 for operational activity.

This is not a likely source of scenarios.

A.2.9 System Information Exchange Matrix (SV-6)

This matrix describes, in tabular form, the physical aspects of how the information exchanges called for in OV-2 actually are (or will be) implemented in terms of protocols, data formats, etc.

This is particularly useful for understanding the potential for overhead and constraints introduced by these choices. See SV1 for system, system elements, system components, system functions. See SV-4 for the inputs to and outputs from the system elements.

Possible scenarios include

- interoperability (from media and format)
- security (from classification)
- performance (from frequency, timeliness, and throughput data)

A.2.10 System Performance Parameters Matrix (SV-7)

This matrix builds on the "system element interface description" (SV-1) by portraying the *current* hardware and software performance characteristics of each system, and the or required performance characteristics at specified times in the *future,* geared towards TV-2. See SV-1 for system, system element, and system component.

This product provides the following data that are needed for attribute-specific analyses:

- mean time to failure (MTTF), maintainability, availability, system initialization time, data transfer rate, and program restart time for platforms
- data throughput/capacity, response time, effectiveness, and mean time between software failures for application software

Possible scenarios include

- dependability (from MTTF for hardware and application software)
- performance (from data transfer rate and throughput/capacity, system initialization, program restart, and response time)

A.2.11 System Evolution Description (SV-8)

This depicts how a suite of systems will be made more modern over time, including evolution and/or migration steps to accommodate the specific information requirements, performance parameters, and technology forecasts provided in other products. See SV-1 for system, system element, and system component.

A possible scenario is modifiability (from upgraded functionality).

A.2.12 System Technology Forecast (SV-9)

This contains predictions about the availability of emerging technologies, specific hardware/ software products, and industry trends in short-, mid-, and long-term intervals, focused on technology areas that are relevant to the architecture's purpose.

A possible scenario is modifiability/interoperability (from upgraded capabilities).

A.2.13 System Activity Sequence and Timing Descriptions (SV-10a, 10b, and 10c)

These descriptions consist of three types of models needed to refine and extend the systems view, to adequately describe the dynamic behavior and performance characteristics of the architecture. The three descriptions are described below:

- Systems rules model (SV-10a) — constraints imposed on system functionality due to some aspect of system design or implementation, expressed in a formal language
- Systems state transition description (SV-10b) — detailed sequencing of functions in a system
- Systems event/trace description (SV-10c) — description of dynamic behavior, tracing the actions in a scenario or critical sequence of events along a given time line (used alone or in conjunction with SV-10b)

This product may reflect system-specific aspects or refinements of critical sequences of events described in the operational view (e.g., performance-critical scenarios). See OV-6a, 6b, and 6c for the sequences of events.

Possible scenarios include

- modifiability (from removing constraints on functionality)
- performance (from execution paths and event times)
- dependability (from points of failure)
- security (from points of attack)

A.2.14 Physical Data Model (SV-11)

Describes how the information represented in OV-7 is actually implemented (that is, how the information exchange requirements actually are implemented and how both data entities and their relationships are maintained in the systems architecture).

Possible scenarios include

- performance (from message size)
- dependability (from file system properties)

A.2.15 Standards Technology Forecast (TV-2)

This consists of detailed descriptions of emerging technology standards and implementing products that are relevant to the systems and business processes covered by the architecture in

short-, medium-, and long-term intervals, with confidence factors for the prediction, along with issues that may affect the architecture. See TV-1 and SV-9 for the timeline of the standards and the prediction of availability of the emerging technologies.

A possible scenario is modifiability/interoperability (from new standards).

A.3 Relationships Between C4ISR Products

[Source: product attribute tables in Appendix A of the C4ISR Framework.]

Product	Gets details from	Implied entities, attributes & relationships
Overview and Summary Information (AV-1)	OV-1 • Organization	
Integrated Dictionary (AV-2)		
High-Level Operational Concept Graphic (OV-1)	OV-3 • Operational Information Element	
Operational Node Connectivity Description (OV-2)		OV-3 • Operational Information Element OV-5 • Activity
Operational Information Exchange Matrix (OV-3)	OV-2 • Need line • Operational Node OV-5 • Activity	
System Interface Description (SV-1)	SV-2 • Communication Node	OV-2 • Need line
Technical Architecture Profile (TV-1)		

Table A-1: Detailed Observations About Essential Products

Product	Gets details from	Implied entities, attributes & relationships
Command Relationships Chart (OV-4)	OV-1 • Organization	
Activity Model (OV-5)	OV-2 • Operational Node	
Operational Activity Sequence and Timing Descriptions (OV-6a, 6b, and 6c)		OV-2 • Operational Node
Logical Data Model (OV-7)		
Systems Communications Description (SV-2)	SV-1 • Systems Node • System • Link	OV-2 • Need line • Operational Node SV-1 • Systems node contains system • Operational node maps to systems node • Link implements need line
Systems Matrix (SV-3)	SV-1 • System SV-6 • System Information Element	
Systems Functionality Description (SV-4)	SV-1 • Systems Function • Systems Node	SV-6 • System Information Element
Operational Activity to System Function Traceability Matrix (SV-5)	SV-1 • System Function OV-5 • Operational Activity • System Function	

Table A-2: Detailed Observations About Supporting Products

Product	Gets details from	Implied entities, attributes & relationships
System Information Exchange Matrix (SV-6)		SV-1 • System, System Element • Application Software (System Component) • System Function • System performs system function • System element performs system function • System information element is input to system function • System is source of system information element
System Performance Parameters Matrix (SV-7)		SV-1 • System, System Element • Application Software (System Component) • System contains system element • System element contains system component
System Evolution Description (SV-8)	SV-1 • System • System Element • System Component	
System Technology Forecast (SV-9)		
System Activity Sequence and Timing Descriptions (SV-10a, 10b, and 10c)		SV-1 • Systems Node
Physical Data Model (SV-11)		
Standards Technology Forecast (TV-2)		TV-1 • Reference Model • Service Area, Service • Reference model includes service area • Service area includes service

Table A-2: Detailed Observations About Supporting Products (Continued)

Appendix B: Screening Questions for the C4ISR Operational View

B.1 Operational Environment

B.1.1 Mission and Geography

- Which missions in which geographical regions would you expect to place stress on the system resources?
- How would you define the workload of each such stressful situation?
- What assets would you expect to be deployed?

When these have been answered and follow-up questioning and explanations have been completed, we should have an operational picture of the expected stressful situation. Presumably the architect would use the operational views to represent this picture. This will help us to understand what is going on.

B.1.2 Organization and Command Structure

- What is the command structure for the mission/geographical region?
- How does this command structure interweave with the U.S. Coast Guard (USCG) operational hierarchy?
- Can this structure change with asset or communications failure?

B.1.3 Quality Attributes

B.1.3.1 Dependability

- When the assets are deployed on a mission, are there different ways in which the communications system can fail (for example, going out of line of site, equipment failure)?
- Are their redundant communication paths, and how is reconfiguration achieved on failure?
- Are their particular assets whose failure would cause the mission to be degraded or abandoned?

- What types of fixes to failed equipment are done during the mission? Are there trained personnel and spare parts on board?
- Are there cases where faults can propagate causing widespread failures?

B.1.3.2 Performance

Bandwidth

- How much information will be transmitted over the communications paths and at what intervals?
- Are there bursts of information that could saturate the communications paths and cause delays to critical information?
- Are there large messages being transmitted, which could interfere with other traffic (e.g., initialization, charts)?

Response Times

- Do operational personnel require some critical responses, and what type of delay is acceptable? Have you performed an analysis to ensure that this can be met under high-stress burst conditions within reason?
- Are there some critical response times to detected trigger events?

B.1.3.3 Security

- Are there missions where the enemy will be attempting to create false alarms to confuse the mission? How can you prevent these false alarms from affecting the mission?
- Is there critical information that must be consistent and have protected access against unauthorized disclosure?
- Is multi-level access necessary?
- Is denial of service an issue?

B.2 Logistics Environment

B.2.1 Platform and Command Center Upgrades

- Which anticipated platform upgrades will be most challenging for team formation and mission operation?
- Which asset upgrades will happen together (all operational assets of Type A will be removed from service, upgraded, and restored to service together), and which will be done on a one-at-a-time basis?

- Will different types of upgrades have different strategies (e.g., new functionality, technology refreshment, problem fixes)?

B.2.2 Evolution Strategy

- Is there a policy about keeping technology relatively fresh?
- Will components be sustained beyond their end of life?
- Is there a policy about installing obsolete technology?

B.2.3 Legacy Systems

- Are legacy systems to be upgraded as part of the plan?
- How will the legacy system upgrades be accomplished?

B.2.4 Quality Attributes

B.2.4.1 Modifiability

- Is there a list of expected changes that will occur to the system during its life expectancy (e.g., threats, technology, operational, budget, schedule)?
- How have these predicted changes affected the system architecture?
- Have contingency plans been developed based on these predicted changes?
- Are the requirements likely to remain stable or to change intrusively?

B.2.4.2 Inter-operability

- Are there well-defined procedures for developing interfaces to external agencies?
- Are all such interfaces defined by some standard?
- Is there a plan to configuration manage the interfaces?
- Is there a common data model for interfaces shared by all participating agencies?

The above questions could also be important for different integrated deep-water system (IDS) assets.

Appendix C: Example ATAM Scenarios

The following example scenarios were generated during actual ATAM evaluations.

C.1 Scenarios About the Development Project

Scenario [Stakeholder(s); Quality/Qualities]

Decrease cost of development by 20%. [Customer, project manager; cost, ability to subset]

Project manager adds two new developers in a way to make them productive quickly. [Developer, project manager; buildability, conceptual integrity]

Implement a skeletal version of the system first. Add functionality incrementally. [Project manager, developers, integrators, testers; ability to subset]

An error is discovered during testing. The development organization identifies the source of the error and corrects it. [Developer, maintainer; maintainability, testability]

Schedule the threads and processes; verify that the set is schedulable. [Developer, performance engineer; performance, schedulability]

The network is overloaded (for network-based systems). Re-partition tasks to reduce traffic. [Developer, performance engineer; performance, schedulability]

Predict priority, execution time, memory space usage, deadline satisfaction, and resource contention when adding new software. [Developer, performance engineer; schedulability, conceptual integrity]

Produce a version of the system that runs on a desktop machine (for systems that run on specialized platforms or are embedded). [Developers; portability]

C.2 Scenarios About the System During Operation

Scenario [Stakeholder(s); Quality/Qualities]

One of the outputs is erratic. Track down and fix the bug. [Tester, maintainer; testability, modifiability]

An error (e.g., divide by zero) occurs during processing. How does the system detect it, report it, handle it, and/or correct it? [User, tester, developer, maintainer, field support; conceptual integrity, testability]

System initialization and start-up occurs. How does the architecture handle it? Perhaps application code must run before system start-up is complete. [Developer, user, maintainer; conceptual integrity, modifiability]

Allow a user to access the system remotely (e.g., via the Web, via radio signals, via dial-up). [User; modifiability, security]

Verify the system's computations, and/or the timing of those computations, at runtime. [Tester, maintainer, performance engineer; modifiability, fault tolerance, reliability]

One processor or server goes down during operation. Half of the processors or servers go down during operation. The network fails during operation. One or more input devices fail during operation. [User; fault tolerance, availability]

Transmit data (to other systems, or across the network in this system) in encrypted form. [User; security]

If using a distributed system, change the processor on which a specific piece of software runs. [User, maintainer, performance engineer; fault tolerance, scalability, portability]

If using a distributed system, use a fault tolerance scheme that accounts for software migration off of a failed or damaged processor. [Maintainer; modifiability, fault tolerance, availability]

Degraded operation mode: A user wants to continue to operate the system (perform useful work) in the presence of one of the following kinds of failures: database server, application server, or partial network failure. [User, maintainer; fault tolerance, availability, modifiability]

Reduce system start-up (or re-start) time by a given factor. [User, performance engineer; modifiability, performance]

An algorithm fails; system must revert back to using a previous (and more reliable) version. [User; fault tolerance]

Only certain users can access certain data in the database. [User; modifiability, security]

Have the system operate for 24 hours a day, 7 days a week. [User; reliability]

C.3 Scenarios About Maintaining or Upgrading the System

Scenario [Stakeholder(s); Quality/Qualities]

Add a new data server. [Maintainer; modifiability, performance]

Replace one of the input devices with one of similar or slightly enhanced capability. [Maintainer; modifiability]

A new input device replaces five current ones. [Maintainer; modifiability]

Performance requirements are doubled. Data throughput is doubled. [Maintainer, performance engineer; performance, modifiability]

The number of users and/or the number of inputs doubles. [Maintainer, performance engineer; performance, modifiability, scalability]

The requirement to use a particular commercial off-the-shelf (COTS) component is levied. [Maintainer; openness, modifiability]

The computer is replaced. If there is more than one computer, heterogeneous computers are introduced. The compiler is replaced. The operating system is replaced. The network or bus is replaced. [Maintainer; modifiability, portability]

Add a completely new function; compute a new result. [Customer, user; modifiability, conceptual integrity]

Make the system interact with some other (specified) system in some (specified) way. (For example, have the system participate in a cross-system simulation exercise.) [User, users of other system; modifiability]

Introduce a new data format or message format. [User, users of other system; modifiability]

Introduce a recording/playback capability. [User; modifiability]

Replace the middleware or communications infrastructure with one supplied by a different vendor. [Customer, performance engineer; modifiability]

Add automated decision aids. [User; modifiability]

Replace the database. Add an object-oriented database. [Developer, customer; modifiability]

Replace the user interface. [User, developer; modifiability]

Introduce data from a completely new source; introduce real-time data. [User, customer; modifiability, performance]

C.4 Scenarios Specifically About a Product Line Architecture

Scenario [Stakeholder(s); Quality/Qualities]

A product developer wants to build the smallest possible running system (using only the architecture and some subset of the reusable assets of the product line) that computes some useful result. [Product line application developer; applicability of architecture to product line, conceptual integrity, ability to subset]

A new reuser (subscriber) wishes to join the product line, but needs to know if his performance and accuracy requirements can be met using the core assets. [Reuser, (new) product line application developer; applicability of architecture to product line, conceptual integrity, ability to subset]

Issue a new release of the core asset library. [Product line application developers; applicability of architecture to product line]

An error is found in a member of the product line, in the field. How is the error corrected, and how does it affect the core assets and other members of the product line? [Testers, service representatives; ability of architecture to isolate errors]

The domain (scope) of the product line is expanded to include. . . . [Product line manager; marketers; product line application developers; flexibility, applicability of architecture to scope, modifiability]

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	June 1999	Final

4. TITLE AND SUBTITLE
Architecture Tradeoff Analyses of C4ISR Products

RO

5. AUTHOR(S)
Mario Barbacci, William G. Wood

6. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

7. PERFORMING ORGANIZATION REPORT NUMBER
CMU/SEI-99-TR-014

8. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
HQ ESC/DIB
5 Eglin Street
Hanscom AFB, MA 01731-2116

9. SPONSORING/MONITORING AGENCY REPORT NUMBER
ESC-TR-99-014

10. SUPPLEMENTARY NOTES

12.a DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified/Unlimited, DTIC, NTIS

12.b DISTRIBUTION CODE

13. ABSTRACT (maximum 200 words)

Early evaluation of the architecture of a system or a product line of systems is a low-cost risk reduction method for determining whether the system(s) will achieve its business and quality goals. The Architecture Tradeoff Analysis Method (ATAM) is an architecture evaluation technique currently evolving at the Software Engineering Institute (SEI). The input to the ATAM consists of a system or product line architecture and the perspectives of stakeholders involved with that system or product line. The output of the ATAM is (1) a collection of scenarios that help specify the context of the system's or product line's use and the product line's evolution, (2) improved architectural documentation (usually), and (3) analysis results (in particular, a set of issues to consider, risks, and potential sensitivity and tradeoff points within the architecture). Currently, there are no generally accepted industry-wide standards for describing a system architecture, and ATAM evaluations are often tailored to the available documentation.

The Architectures Directorate of the C4I Integration Support Activity (CISA), Office of the Assistant Secretary of Defense for Command, Control, Communications, and Intellegence (OASD[C3I]) has defined a framework for architecture development, presentation, and integration to be used across the military services and defense agencies. This framework for Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance (C4ISR) is becoming the required method for describing information systems within the Department of Defense (DoD) and other U.S. Government agencies. This report describes how various C4ISR products can be used in the context of an ATAM evaluation and their relative value for generating quality attribute-specific scenarios required for an ATAM evaluation.

14. SUBJECT TERMS
architecture evaluation techniques; Architecture Tradeoff Analysis Method (ATAM); Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance (C4ISR); product line architecture; system architecture

15. NUMBER OF PAGES
40

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL